CS230

MR-AC using deep learning

Jonathan Fisher
Department of Electrical engineering
Stanford University
jofisher@stanford.edu

Abstract

Accurate Positron emission tomography (PET) results require correcting for patient's body attenuation. The classic way of using X-ray Computed Tomography (CT), expose the patient to high dosage of radiation. Magnetic Resonance Imaging (MRI) is both safer (no extra radiation) and provide additional and useful soft tissue information. The downside is that evaluating attenuation using MR images is not trivial.
This paper suggests a way to perform image to image translation: MR base attenuation correction (MR-AC) to CT-AC using conditional generative adversarial networks (cGANs) - the state of the art deep learning image-to-image translation method. The paper than examine how the generator performance can be further improved by using FusionNet - a fusion between U-net and ResNet.

1 Introduction

PET is a quantitative technology for imaging metabolic pathways and dynamic processes in vivo. In order to obtain accurate PET images, the emission data recorded during a PET scan must undergo different corrections prior to image reconstruction. One of the correction is an AC.
Hybrid PET/CT systems provide complementary and intrinsically coregistered CT and PET image volumes. The CT transmission data is also used for AC. The translation of the CT transmission data into linear attenuation coefficients (CT-AC) is considered to be simple and accurate.
Due to superior soft tissue contrast offered by magnetic resonance imaging (MRI) and the desire to reduce unnecessary radiation dose, the radiology community's interest to replace CT with MRI has been rapidly growing.
The problem is that it is not trivial to create a reliable MRI base AC (MR-AC). Therefore, methods are needed to derive CT-equivalent representations. MR-AC to CT-AC translation is still a very active field of research.
This project will investigate the use of cGANs to create MR-AC. The input to our algorithm is an MRI image, then cGANs are being used to create an MR-AC image that should be as close as possible to the CT-AC image.

2 Related work

Many different methods have been used to tackle this problem in the past. The methods can be roughly divided into four categories:

- Tissue segmentation - segment the MR image voxels into a discrete set of tissue types, (such as air, fat, soft tissue, and bone) and then assign a different CT number for each tissue type [1]. The biggest problem with this category is that conventional MR image cannot reliably differentiate between bone and air.

- Statistical learning or model fitting techniques that map MR voxel intensities (or intensity patterns) to CT numbers [2]. This approach suffers from the same problem as described above.
- Atlas-based approaches apply image registration to align a target MR image to an atlas MR, the corresponding information can then be used to warp the associated atlas CT image to the target MR to generate a synthetic CT (sCT) [3]. The main challenge is to accurately register patients with large anatomical variations using the same atlas. There are ways to partially overcome these difficulties, like the use of multiple atlases and the design of atlas fusion methods.
- Deep learning mapping approaches like U-net for MR to CT image translation [4].

Recently, a new and promising approach to performing image to image translation using cGANs has been proposed by [5]. The use of cGANs networks as a general-purpose solution has been demonstrated to work well for a verity of different application. cGANs not only use a U-net to learn the mapping from an input image to output image, but also learn a loss function to train this mapping. A problem that traditionally would require investigating the ideal loss formulations. For this reason cGANs significantly outperforms [4].

3 Dataset and Features

The data contains PET/MR and PET/CT scans of a single patient:

- 108 coronal slices of both PET/MR scan (Signa PET/MR: GE healthcare) using a two-point Dixon MR sequence performed approximately 120 minutes after a PET/CT scan (mCT: Siemens healthcare). The CT images at 140 kVp were converted to photon attenuation coefficients at 511 keV. Both MR and CT are 600X600 color images.
- 89 transverse slices of zero-TE (ZTE) MR sequence images as the input images and the corresponding CT images. Both MR and CT are 59X59 color images.

The data were preprocessed to remove artifacts and then augmented using Elastic transformation [6]. I used only two prepossessed images as my dev set (one of each MR sequence), all the rest were used for training (the dev set augmented images were not used).

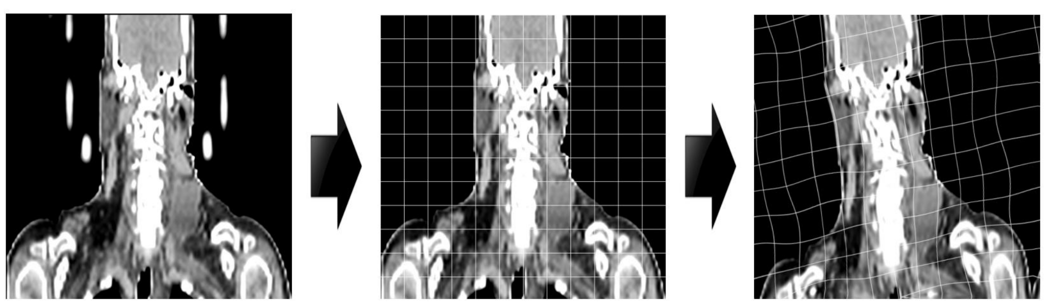

Figure 1: Left: raw data. Middle: prepossessed data, after removing artifacts. Right: augmented data using elastic transformation. (The grid on the middle and right images is only to demonstrate what is elastic transformation, the images that were used doesn't include grid)

All the images were then reshaped to be 256X256 color images.

4 Methods

cGANs architecture contains two components:

1. The generator, $G(x, z)$, is trained to get an MR image as an input and produce a CT image that cannot be distinguished from the real compatible CT image.

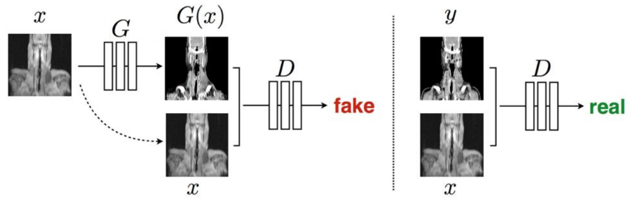

Figure 2: The discriminator, D, learns to classify between fake (synthesized by the generator) and real images. The generator, G,learns to fool the discriminator

2. The discriminator, $D(x, y)$, is trained to identify which couple of CT and MRI images is real, and which is fake (the CT image was produced by the generator).

The algorithm objective:

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[logD(x, y)] + \mathbb{E}_{x,z}[1 - logD(x, G(x, z))]$$

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}||y - G(x, z)||_1$$

$$G^* = \underset{G}{\operatorname{argmin}}\ \underset{D}{\operatorname{argmax}}\ \mathcal{L}_{cGAN}(G, D) + \lambda\mathcal{L}_{L1}(G)$$

In the model, the noise z was in the form of dropout, applied on several layers of the generator at both training and test time.

The generator suggested by [5] uses a "U-net" architecture, an encoder-decoder with skip connections between mirrored layers in the encoder and decoder stacks. This generator architecture was used as a baseline. The improved "FusionNet" generator, inspired by [7], is basically a fusion of U-net and ResNet. The FusionNet generator includes a residual layer in each U-net level. FusionNet is therefor a much deeper network that produces more accurate CT images.

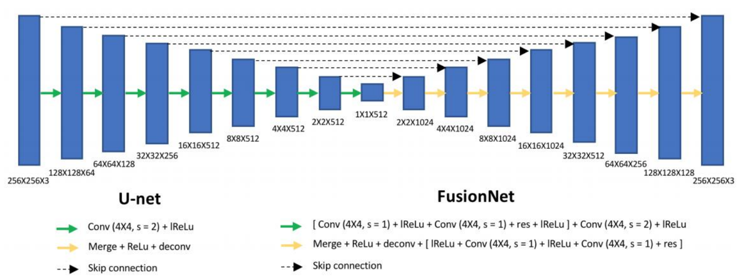

Figure 3: The two Generators architectures

The discriminator architecture is called "PatchGAN". It tries to classify if each NXN patch in an image is real or fake (By running this discriminator convolutionally across the image and averaging all responses to provide the output of D). The assumption is that the discriminator only needs to validate that the high-frequency content is real since L_1 norm force low-frequency correctness.

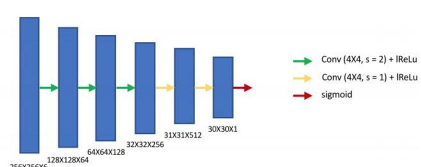

Figure 4: The Discriminator architecture

Networks optimization is being done by alternating between one gradient descent step on D, then one step on G. Rather than training G using the saturated cost the algorithm uses the unsaturated cost, as mentioned in class and in the original GAN paper.

5 Experiments/Results/Discussion

To evaluate the performance I used L_1, L_2 normalized norms:

$$L_1 = \frac{\sum_{slices} \sum_{pixels} |O_{i,j}^{<s>} - T_{i,j}^{<s>}|}{\#slices \cdot \#pixels}$$

$$L_2 = \sqrt{\frac{\sum_{slices} \sum_{pixels} (O_{i,j}^{<s>} - T_{i,j}^{<s>})^2}{\#slices \cdot \#pixels}}$$

(s indicate the slice/image number and i, j indicate the pixel in the slice)

At first I used the default [5] parameters to optimize the network:
I used minibatch SGD and applied Adam solver ($\alpha = 0.0002, \beta_1 = 0.5, \beta_2 = 0.999$).
The dropout rate was 0.5 for the 3 first connections of the U-net generator's decoder (3 most left yellow arrows in Figure 3: $1X1X512 \rightarrow 2X2X1024, 2X2X1024 \rightarrow 4X4X1024, 4X4X1024 \rightarrow 8X8X1024$)

I tested the performance on [5] default network and parameters. The initial results were quite impressive.
The next step was to see how much the deeper generator can improve those results. Indeed, as expected, the "FusionNet" generator was able to give much better results using the default parameters.
Next, I tried to perform hyperparameter tuning on the FusionNet. Due to time limitation, I tuned the network using only on the reprocessed data (didn't use the augmented data) for only 50 epochs. Even then the training time was quite long so I focused my efforts on finding optimal dropout rate, where every iteration took about an hour and a half. After almost a day of tests, I couldn't find a better dropout rate than the default one.
The final step was to see how using the augmented data improve the results I got.
By augmenting the data only once sing (double the pictures) I was able to further improve the network performance.
Since my dataset is small I wanted to verify that I don't overfit to the training set. Initially, I took the average over all the slices in the training set and compared the dev set and training set L_1, L_2 norms. The problem with this comparison is that the dev set image's power is much greater than average training set image's power (many images are mostly black). A better way to check for overfitting is to compare to training slices with similar power, for example, the training slices that are adjacent to the dev set slices.

Generator architecture, train data	L_1 norm (train)	L_2 norm (train)	L_1 norm (train neig.)	L_2 norm (train neig.)	L_1 norm (dev)	L_2 norm (dev)
U-net, prepossessed data	14.266	31.987	17.044	33.83	18.291	36.24
FusionNet, prepossessed data	13.984	30.644	17.93	35.089	16.276	33.363
FusionNet, prepossessed and augmented data	11.432	26.051	15.672	30.412	**15.098**	**30.6**

Table 1: The Normalized L_1, L_2 norms on the entire train set, train set slices that are adjacent to the dev set slices (train neig.), and dev set.

From the table we can learn that the network doesn't overfit the data to the training set.

Some qualitative results: generator prediction on the dev set (after training FusionNet on the prepossessed + augmented data):

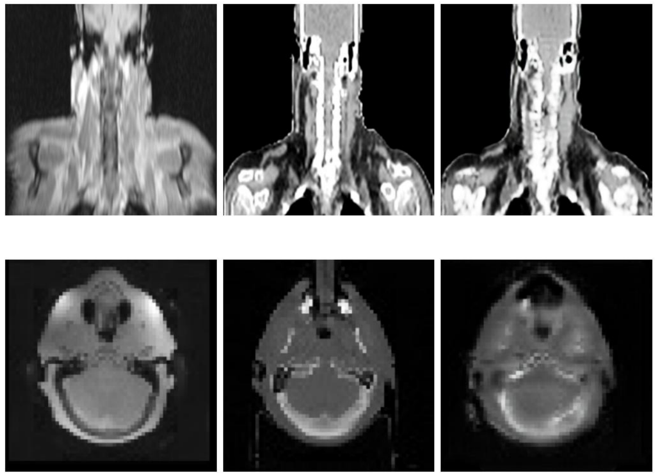

Figure 5: Left: MRI image. Middle: CT based attenuation. Right: Generated attenuation image

6 Conclusion/Future Work

[5] is the state of the art method used for image-to-image translation and indeed was a great baseline solution for this problem. By using FusionNet architecture, a deeper U-net, I was able to further improve the generated results.
It's not always easy to find suitable or big enough datasets, especially when dealing with medical data. Using elastic transformations for data augmentation helped alleviate the problem and improve algorithm performance. That being said, it's important to test the algorithm performance on a new patient.
Due to time limitations and since model training is very time consuming (even when using a GPU), hyperparameter optimization efforts were limited. If I had more time to tune the hyperparameters I'm pretty sure that I could get better results.
I found an additional great dataset at the beginning of the quarter that was perfect for this project. For some reason, I didn't get access to it on time. I'm looking forward to getting it and test the algorithm perform on a different patients.

References

[1] Franke J. Salomon A. Palmowski M. Donker H.C.W. Temur Y. Mottaghy F.M. Kuhl C. Izquierdo-Garcia D. Fayad Z.A. Kiessling F. Schulz V. Berker, Y. MRI-Based Attenuation Correction for Hybrid PET/MRI Systems: A 4-Class Tissue Segmentation Technique Using a Combined Ultrashort-Echo-Time/Dixon MRI Sequence. *Journal of Nuclear Medicine*, 53(5):796–804, 2012.

[2] M. Kapanen and M. Tenhunen. T1/T2*-weighted MRI provides clinically relevant pseudo-CT density data for the pelvic bones in MRI-only based radiotherapy treatment planning. *Acta Oncologica*, 52(3):612–618, 2013.

[3] N. Koutsouvelis M. Rouzaud R. Miralbell H, Arabi and H. Zaidi. Atlas-guided generation of pseudo-CT images for MRI-only and hybrid PET–MRI-guided radiotherapy treatment planning. *Physics in Medicine & Biology*, 2016.

[4] X Han. MR-based synthetic CT generation using a deep convolutional neural network method:. *Medical Physics*, 44(4):1408–1419, 2017.

[5] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-Image Translation with Conditional Adversarial Networks. *arxiv*, 2016.

[6] Bruno G. do Amaral. Elastic transform for data augmentation, 2016.

[7] Tran Minh Quan, David G. C. Hildebrand, and Won-Ki Jeong. Fusionnet: A deep fully residual convolutional neural network for image segmentation in connectomics. *CoRR*, abs/1612.05360, 2016.